



Tipo entero

Transcripción

[00:00] Hola a todos. ¿Qué tal? Sean bienvenidos a una parte más de su curso Primeros pasos en Java. Como podrán ver ya cambio de computador, estoy en una Mac. Estoy usando el mismo Eclipse que ya hemos instalado anteriormente. No se preocupen, es el mismo software, es el mismo programa que estamos usando.

[00:20] Bueno, a modo de repaso anterior de lo que hemos hecho en la última clase, vamos a crear rapidito un nuevo proyecto, Java primeros pasos, y le damos a finish. Como ya hemos visto, el Package Explorer view nos va a proporcionar la vista SRC y en este caso no vamos a poder ver la carpeta bin, que ya vimos en el anterior video, porque es una carpeta oculta en esta vista de Eclipse.

[00:56] Si nosotros necesitáramos ver lo que está dentro de bin, deberíamos entrar por la vista navigator o por el explorador de Windows en todo caso. Entonces, vamos a comenzar aquí creando una nueva clase. Estamos aquí y vamos a llamar a nuestra clase TipoVariable, porque vamos a ver tipos de variables en esta oportunidad.

[01:32] No marcamos aquí la opción para marcar el método Main, no deseamos generar automáticamente, lo vamos a escribir nosotros. Le damos finish. Y bien, ya tenemos nuestra clase TipoVariable creada. Entonces, para recordar un poco lo que hicimos en la clase anterior, la idea es volver a imprimir un valor en la consola en este momento.

[02:00] Si recordamos bien, no podemos ejecutar directamente la instrucción de imprimir en la consola fuera del contexto de un método, entonces vamos a crear nuestro método Main, que ya dijimos que es el método principal que necesita nuestra clase. Comenzamos a escribir: `public void main`, es `public static`. No se preocupen ahora por entender qué significa cada una de estas palabras.

[02:26] Ya poco a poco lo van a entender en su momento. Y aquí le pasamos los argumentos. Perfecto. Yo hice la letra aquí un poco más grande para que puedan visualizar en dispositivos pequeños. Normalmente van a usar un tipo de letra un poco más pequeño en Eclipse, pero no encuentren extraño esto, yo aumenté el tamaño fuente a propósito.

[02:52] Y bueno, entonces aquí vamos a ejecutar `System.out.println`, y le ponemos nuestro valor "Hola Mundo". Salvamos apretando el ícono de aquí o haciendo "Ctrl + S" directamente, es un atajo. Y en el botón verde de aquí le damos para correr la clase y listo, tenemos "Hola Mundo" en la consola.

[03:18] Ahora, el objetivo de esto es ir un par de pasos más adelante, entonces ya vamos a comenzar a trabajar con variables, un tipo de programación más dinámica. ¿Para qué nos pueden servir variables? Para asignar valor a ellas e ir las utilizando a lo largo de nuestro programa.

[03:35] Entonces, por ejemplo si yo quiero si yo quiero declarar una variable edad y quiero asignarle mi edad, en este caso vemos que tenemos un error aquí al costado de edad. ¿Por qué? ¿Qué nos dice el compilador? Que edad no puede ser resuelto a una variable. ¿Por qué? Porque Java es un lenguaje fuertemente tipado.

[03:54] ¿Qué significa esto? Que cada variable que nosotros declaremos tiene que tener un tipo de dato asignado para que el lenguaje entienda a: voy a trabajar con números, voy a trabajar con strings o el tipo de dato que estemos utilizando. Para números, Java nos provee un tipo de variable que se llama `int`.

[04:17] Este tipo de variable es una palabra reservada de Java que va a especificar que vamos a guardar un valor de tipo entero en esta variable.

Ahora, cada palabra que está en este color morado, violeta, color vino, que está aquí en el programa, es una palabra reservada de Java. ¿Qué significa esto? Que nosotros no podemos usar estas palabras en nombres de métodos o nombres de variables.

[04:44] Por ejemplo si yo quiero llamar `int static`, él no me va a dejar usar este nombre de variable porque ya es una palabra reservada del lenguaje de programación. Entonces, volvemos a `edad` y salvamos, compilamos y corremos el programa y tenemos un "Hola Mundo" de nuevo. Pero prácticamente no estamos utilizando esta variable.

[05:07] Lo que vamos a hacer ahorita es imprimir esta variable para darle un uso, y ver que en efecto este valor está siendo asignado a `edad`. Entonces, volvemos a `System.out.println`, listo. Y aquí viene algo curioso. Si nosotros asignamos `edad` dentro de las comillas, vamos a darnos cuenta, salvamos otra vez, damos a correr, que él imprime `edad`. ¿Por qué?

[05:40] Porque le estamos pasando como parámetro un string, una palabra, y no la variable en sí. ¿Cómo resolvemos esto? Le pasamos directamente el nombre de nuestra variable como parámetro al método `println`, salvamos nuevamente, ejecutamos y tenemos aquí 28. Perfecto. ¿Eso es lo único que podemos hacer con las variables? No. ¿Por qué?

[06:06] Porque a `edad` podemos sobrescribirle el valor, ahora por ejemplo a 47, copiamos nuevamente esta línea, salvamos y vemos que `edad` ya no vale 28, ahora vale 47. ¿Por qué? Porque hemos asignado un nuevo valor para ella. De esa misma forma podemos asignarle valores aritméticos como por ejemplo `46 + 88`, por ejemplo, es un número que se me ha ocurrido ahora al azar.

[06:41] Hacemos esto, salvamos antes, y tenemos que él resolvió la suma dentro de la variable. Entonces, ¿será que solo podemos hacer esto con el

System.out.print? Vamos a ver un truco más con esto, porque el signo + tiene un uso que no solamente se aplica para números en el caso de Java.

[07:07] Por ejemplo, si nosotros queremos imprimir algo así como "mi edad es" y queremos usar la variable edad, lo único que tenemos que hacer es agregar el signo +. ¿Por qué? Porque el signo +, si es usado entre valores numéricos, él va a ejecutar una operación numérica entre esos valores, pero si es utilizado como string, él no va a hacer una operación de suma, él va a hacer una operación de concatenación de valores.

[07:47] Entonces, lo que él va a hacer aquí es concatenar "mi edad es", esta oración, con el valor ya calculado aquí de edad. Él va a interpretar todo esto como un string solo. Vamos a comprobarlo. Salvamos nuevamente, le damos play y tenemos "mi edad es 134". Vamos a darle un espacio para una mejor formatación y listo, "mi edad es 134".

[08:19] Así sabemos ya cómo es que funciona la concatenación de strings con números, sabemos que podemos realizar operaciones aritméticas dentro de la variable y asignar ese resultado directamente a la variable y aprendimos que podemos sobrescribir el valor de la variable cuantas veces lo consideremos necesario.